

## CLAIMS

- 1     1.     A method for separating data blocks referenced by a writable virtual disk (vdisk)  
2     from data blocks referenced only by a backing store of a storage system, the method com-  
3     prising the steps of:  
4         loading blocks of the writable vdisk from a disk into a memory, the loaded blocks  
5     including a writable vdisk indirect block having a plurality of fields, each field storing a  
6     valid pointer to a data block or an invalid pointer representing a hole;  
7         loading blocks of the backing store from a disk into the memory, the loaded  
8     blocks including a backing store indirect block having a plurality of fields, each backing  
9     store indirect block field corresponding to a field of the writable vdisk indirect block, one  
10    or more backing store indirect block fields having a pointer to a data block;  
11         searching each field of the writable vdisk indirect block for a hole; and  
12         replacing each field having a hole in the writable vdisk indirect block with a new  
13    pointer to the data block referenced by the corresponding backing store indirect block  
14    field.
- 1     2.     The method of claim 1 wherein the step of replacing comprises the step of:  
2         dirtying the data block pointed to by the backing store indirect block to enable  
3     write allocation of the dirty data block without altering a data content of the data block.
- 1     3.     The method of claim 1 wherein the step of replacing further comprises the steps of:  
2         choosing a new pointer for a newly allocated data block containing the unaltered  
3     data content;  
4         setting bits in block allocation structures for the newly allocated data block; and  
5         placing the new pointer to the newly allocated data block into the field of the wri-  
6     table vdisk indirect block to replace the hole.

- 1    4. The method of claim 3 further comprising the step of:  
2        freeing the dirty data block; and  
3        writing the newly allocated data block to disk.
  
- 1    5. The method of claim 4 further comprising the step of:  
      releasing an association of the writable vdisk to the backing store to thereby separate the writable vdisk data blocks from the backing store data blocks.
  
- 1    6. The method of claim 1 wherein the pointers contained in the writable vdisk indirect  
2    block fields and the backing store indirect block fields comprise logical volume block  
3    numbers (VBNs).
  
- 1    7. The method of claim 1 wherein the invalid pointers contained in the writable vdisk indirect  
2    block fields comprise a zero logical volume block number (VBN).
  
- 1    8. The method of claim 1 wherein the plurality of fields in the writable vdisk indirect  
2    block are a writable vdisk level 1 buffer and the plurality of fields in the backing store  
3    indirect block are a backing store level 1 buffer.
  
- 1    9. An apparatus for separating data blocks referenced by a writable virtual disk (vdisk)  
2    from data blocks referenced only by a backing store of a storage system, the apparatus,  
3    comprising:  
4        a backdoor message handler adapted to load blocks of the writable vdisk and  
5        backing store from disk into a memory of the storage system;  
6        a writable vdisk indirect block in the memory having a plurality of fields, each  
7        field storing a valid pointer to a data block or an invalid pointer representing a hole;

8           a backing store indirect block in the memory having a plurality of fields, each  
9 backing store indirect block field corresponding to a field of the writable vdisk indirect  
10 block, each backing store indirect block field having a pointer to a data block;  
11           a special loading function for searching each field of the writable vdisk indirect  
12 block for one or more fields representing a hole; and  
13           a write allocator for replacing each field representing a hole in the writable vdisk  
14 indirect block with a new pointer to the data referenced by the corresponding backing  
15 store indirect block field.

1    10. The apparatus of claim 9 wherein the write allocator is further adapted to:  
2           choose a new pointer for a newly allocated data block containing an unaltered data  
3 content, set bits in block allocation structures for the newly allocated data block, and  
4 place the new pointer to the newly allocated data block into the field of the writable vdisk  
5 indirect block to replace the hole.

1    11. The apparatus of claim 10 wherein the write allocator is further adapted to:  
2           free the dirty data block and write the newly allocated data block to disk.

1    12. The apparatus of claim 9 wherein the backdoor message handler loads the blocks of  
2 the writable vdisk and the blocks of the backing store during periods of reduced process-  
3 ing activity.

1    13. The apparatus of claim 9 wherein the pointers contained in the writable vdisk indirect  
2 block fields and the backing store indirect block fields comprise logical volume block  
3 numbers (VBNs).

4    14. The apparatus of claim 9 wherein the invalid pointers contained in the writable vdisk  
5 indirect block fields comprise a zero logical volume block number (VBN).

1 15. The apparatus of claim 9 wherein the plurality of fields in the writable vdisk indirect  
2 block comprises a writable vdisk level 1 buffer and the plurality of fields in the backing  
3 store indirect block comprises a backing store level 1 buffer.

1 16. A method for operating a storage system that services access requests to data stored in  
2 data blocks on a storage device, the method comprising;  
3 generating a read-only backing store of an organization of data blocks;  
4 generating a writable image of the organization of data blocks, the writable image  
5 including references to the backing store;  
6 separating the backing store and the writable image;  
7 deleting the backing store without interrupting the servicing of the access re-  
8 quests.

1 17. The method of claim 16 wherein the step of separating further comprises the steps of:  
2 searching a plurality of fields of the writable image for indications to reference  
3 the backing store;  
4 replacing each indication with a pointer to a newly allocated data block associated  
5 with the writable image.

1 18. The method of claim 16 wherein the indications to reference the backing store are in-  
2 valid pointer values.

3 19. An apparatus for separating data blocks referenced by a writable virtual disk (vdisk)  
4 from data blocks referenced only by a backing store of a storage system, comprising:

5 means for loading blocks of the writable vdisk from a disk into a memory, the  
6 loaded blocks including a writable vdisk indirect block having a plurality of fields, each  
7 field storing a valid pointer to a data block or an invalid pointer representing a hole;

8 means for loading blocks of the backing store from a disk into the memory, the  
9 loaded blocks including a backing store indirect block having a plurality of fields, each  
10 backing store indirect block field corresponding to a field of the writable vdisk indirect  
11 block, one or more backing store indirect block fields having a pointer to a data block;

12 means for searching each field of the writable vdisk indirect block for a hole; and

13 means for replacing each field having a hole in the writable vdisk indirect block  
14 with a new pointer to the data block referenced by the corresponding backing store indi-  
15 rect block field.

16

1 20. A computer readable medium, including program instructions executing on a com-  
2 puter, the program instructions including instructions for performing the steps of:

3 loading blocks of the writable vdisk from a disk into a memory, the loaded blocks  
4 including a writable vdisk indirect block having a plurality of fields, each field storing a  
5 valid pointer to a data block or an invalid pointer representing a hole;

6 loading blocks of the backing store from a disk into the memory, the loaded  
7 blocks including a backing store indirect block having a plurality of fields, each backing  
8 store indirect block field corresponding to a field of the writable vdisk indirect block, one  
9 or more backing store indirect block fields having a pointer to a data block;

10 searching each field of the writable vdisk indirect block for a hole; and

11 replacing each field having a hole in the writable vdisk indirect block with a new  
12 pointer to the data block referenced by the corresponding backing store indirect block  
13 field.